An upper bound for the total number of options to implement an SDR multi-standard system

Patricia Kaiser, Amine El Sahili, Yves Louët

An upper bound for the total number of options to implement an SDR multi-standard system

Patricia KAISER
Lebanese University and
Supelec Rennes
Email: patricia.kaiser@supelec.fr

Amine EL SAHILI
Lebanese University
Hadath campus - Lebanon
Email: sahili@ul.edu.lb

Yves LOUET
IETR/Supelec, Campus de Rennes
Avenue de la Boulaie, CS 47601, France
Email: yves.louet@supelec.fr

***Abstract*—The design of future multi-standard systems is very challenging. Flexible architectures that exploit common aspects between the different set of standards cohabiting in the device offer promising solutions. In this paper, we adopt the theory of graphs and particularly the study of directed hypergraphs, which helps in providing a theoretical view of the graph structure of the multi-standard system. This graph description exhibits all the alternatives capable of implementing the system. On the other hand, we further provide in this paper an exponential upper bound for the total number of options which are capable of implementing a multi-standard design.**

***Index Terms*—Directed hypergraphs, SDR multi-standard systems, cost function, exponential upper bound.**

I. Introduction

Radio system designers focus their attention nowadays on developing flexible equipments that support a larger number of air interface standards in order to cope with the daily accelerating rate of technology innovation. The Software-defined Radio (SDR) concept [1] was emerging as a potential and efficient solution for making wireless networks highly adaptable and flexible. Design of SDR equipments is very challenging because it is very difficult to design a system that preserves most of the properties of the ideal software radio while being realizable with current-day technology.
The fields of research that aim at solving or at least improving the methods and technologies implicated in the SDR area are numerous. Nevertheless, the SDR community is particularly driving the activities focusing on reconfiguration and reconfigurability [2] in a heterogeneous context [3].

The possibilities to design software radio architectures range from the "Velcro" approach to the "Very Fine Grain" approach. The former approach aims to support several communication standards through dedicated self-contained complex communication components, while the latter is based on manipulating small size operators to support different standards. However, a promising approach to realize an SDR multi-standard terminal is to identify the appropriate common functions and operators between and inside the standards. This is what's called the "parametrization" approach [4]. This last approach aims at designing multi-standard systems made of certain operators (or functions) whose operations can be modified by a simple parameter adjustment. This approach can be extended to lower level entities called common operators [5]. It's a very promising approach consisting of designing radio systems entities in a way that permits to take advantage of the programmable or at least reconfigurable capabilities of the underlying hardware of SDR systems.

In order to explore the different possibilities of an SDR design, it was necessary to describe the interrelationships between the different components in the system. Thus a graph structure representing the decomposition, into less and less complex operators, of each standard which has to be realized in the design was illustrated in [6], [7]. This graph representation provides all the options of implementation capable of realizing the multi-standard system for the designer, who can select the option that meets his demands. However, a cost function which calculates the cost of each of these options is introduced in [5]. This cost function combines both the flexibility and efficiency aspects. In this paper, we will exploit the theory of graphs in this context.

Graph theory is rapidly moving into the mainstream of mathematics mainly because of its diverse applications in different fields. Graph theory [8] is the study of graphs used to model pairwise relations between objects from a certain collection. A "graph" in this context refers to a collection of vertices and a collection of edges that connect pairs of vertices. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edge may be directed from one vertex to another in which case it is called a digraph. Hypergraphs and directed hypergraphs [9] are generalizations of graphs and digraphs respectively.

Our final objective is to optimize the cost function to its minimum value possible and thus solving the optimization problem that finds balance between flexibility and computing efficiency. This will enable us to extract the most appropriate Common Operators (COs) from the most convenient granularity levels leading to the construction of an optimal SDR multi-standard design which takes advantage of the common aspects in use.
One issue towards solving our optimization problem is to try to explore the total number of options which are capable of implementing a multi-standard design, from which one with

a minimum cost is supposed to be extracted. In this paper, we detail this point and work on finding out an upper bound for the total number of the options of implementation of the multi-standard system.

The rest of this paper is organized as follows. After the present section, some basic definitions of directed hypergraphs essential for our work are reported as well as a theoretical description of the graph structure of the SDR multi-standard system as a directed hypergraph. Section 3 briefly introduces a cost function suggested in [5], which can calculate the cost of each possible option of implementation. In the subsequent section, an exponential upper bound for the total number of options which are capable of realizing the multi-standard design is derived. Finally, a conclusion's section ends this paper.

II. Representation for SDR system implementation using graph theory

In this section, we first explore a model for multi-standard systems as a graph with different levels of granularity which enables to select the convenient level depending on each designer's needs. Then, we introduce some required definitions related to directed hypergraphs that will be key notations for the rest of this article. All this was indispensable in order to provide a theoretical representation of the graph structure of the multi-standard system as a directed hypergraph, which is finally presented in the section.

A. Graph modeling of SDR systems

An SDR multi-standard system is represented as a graph in [6]. To model such systems as a graph, it is necessary to distinguish between the different dependencies of the nodes of different levels. A node of a higher level, called a parent node, may have dependencies with nodes of underlying levels, called descendant nodes. An "OR" dependency (left part of fig. 1) means that only one of the descendant nodes (B or C) called several specific times is necessary to implement the parent node (A). However, an "AND" dependency (right part of fig. 1), signifies that all descendant nodes via the "AND" dependency (B & C) are needed to implement the parent node (A) accompanied with certain number of calls.

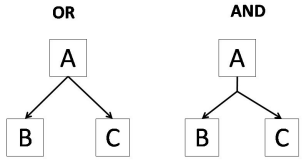

Fig. 1. "OR" and "AND" dependency

A multi-standard reconfigurable system is modeled as a graph with several layers as shown in fig. 2. The roots of this graph, at the top level, represent the standards to be supported by the radio (Wifi and UMTS in the case of fig. 2). Each processing element (PE) occupies a certain layer depending upon its granularity level, where more complex PEs have higher granularity levels than less complex ones that can form their functionalities. Each node in the graph represents an elementary PE. In order to perform the functionalities of this PE, it can be installed by itself in the design, as a unified non divisible block, or it can be realized by some lower-level building blocks. The goal of this approach is to provide the options to the designer, to select a set of operators each of which occupies a certain level of granularity, as dictated by his needs.

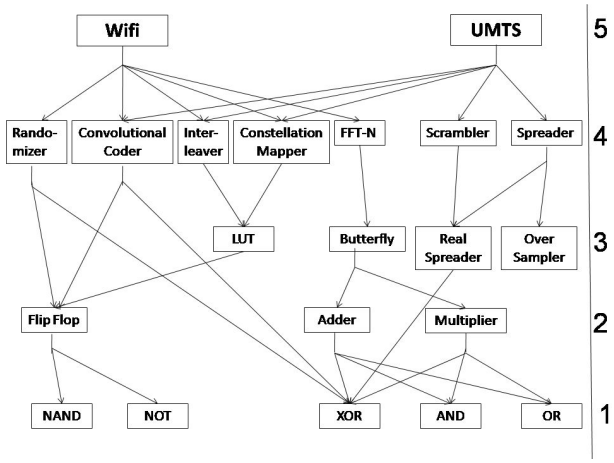

Fig. 2. Global structure of a multi-standard graph (supporting Wifi and UMTS) - transmitter side, including the levels of the blocks on the right side of the figure

B. Basic definitions related to directed hypergraphs

1) Hypergraphs: A hypergraph H is defined by a pair $H = (X, E)$, where $X = \{x_1, x_2, \cdots, x_n\}$ is the set of vertices of H and $E = \{e_1, e_2, \cdots, e_m\}$, with $e_i \subseteq V$, $e_i \neq \phi$ for $i = 1, 2, \cdots, m$, denotes the set of hyperedges of H. Clearly, when $|e_i| = 2 \ \forall \ i = 1, 2, \cdots, m$, the hypergraph is a standard graph.

2) Directed Hypergraphs [9]: A directed hypergraph is a hypergraph but with directed hyperedges (also called hyperarcs) where a directed hyperedge e is an ordered pair $e = (X, Y)$ of (possibly empty) disjoint subsets of vertices; X is the tail of e denoted by $T(e)$ and Y is its head denoted by $H(e)$ [9]. Fig. 3 is an example of a directed hypergraph $H = (X, E)$ where:

$X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}$
$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$ where:
$e_1 = (\{x_1, x_2\}, \{x_3\})$
$e_2 = (\{x_3\}, \{x_7\})$
$e_3 = (\{x_3, x_4\}, \{x_5, x_6\})$
$e_4 = (\{x_7\}, \{x_1\})$
$e_5 = (\{x_1, x_7\}, \{x_8\})$
$e_6 = (\{x_9\}, \{x_8\})$

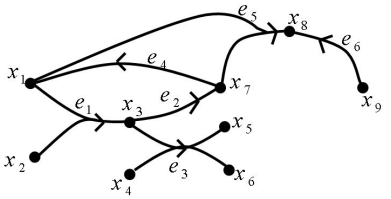

Fig. 3. A directed hypergraph $H = (X, E)$

3) Forward and Backward star [9]: Let $v \in V$. The Forward Star and the Backward Star of node v are defined by:
$FS(v) = \{e_i \in E, v \in T(e_i)\}$ and
$BS(v) = \{e_i \in E, v \in H(e_i)\}$ respectively.
Example In Fig. 3 we have:
$FS(x_3) = \{e_2, e_3\}$ *and* $BS(x_3) = \{e_1\}$.
$FS(x_8) = \phi$ *and* $BS(x_8) = \{e_5, e_6\}$.

4) BF-reductions of hyperarcs [9]: Let $e = (T(e), H(e))$ be a hyperarc in a directed hypergraph H. A BF-reduction of e is a hyperarc $(\{x\}, \{y\})$ where $x \in T(e)$ and $y \in H(e)$.
Example: we have $(\{x_3\}, \{x_5\})$, $(\{x_3\}, \{x_6\})$, $(\{x_4\}, \{x_5\})$, $(\{x_4\}, \{x_6\})$ are all BF-reductions of the hyperarc E_3 in Fig.3.

C. A mathematical model of the graph structure of the SDR multi-standard system

A theoretical representation of the graph structure presented in subsection II-A can be concluded. Formally speaking, the graph structure of a multi-standard system can be viewed as a directed hypergraph defined by the couple (V, E). The set of vertices V will include the blocks (functions and operators) present in the graph structure (example V = {wifi, UMTS, randomizer, convolutional coder, $\cdots$, or} in the graph structure of Fig. 2) and a directed hyperedge $e \in E$ will include the parent node as a tail node while all the necessary descendent node(s) capable of performing its task will form the head node(s) of e. So this means that whenever we have an "AND" dependency, the hyperarc is formed such that the parent node is the tail node and all the descendent nodes via this "AND" dependency are its head nodes. Whereas when faced with an "OR" dependency, the hyperarc will have the parent node as the tail node and only one of its descendent nodes (if more than one exists) via the corresponding "OR" dependency will be the head node. In this way, we form the set of hyperarcs E including all the "OR" and "AND" dependencies present in the corresponding graph structure. For instance, we have ({randomizer},{Flip Flop,XOR}) and ({interleaver},{LUT}) belong to the set of hyperarcs E of the directed hypergraph of fig. 2.

III. The cost function

The graph structure of the multi-standard system (introduced in the previous section) provides all the options capable of implementing the standards to be supported. However, the cost function (CF) yields the cost imposed by any one of these options of implementation.

A. A Cost Function expression

Three key parameters enter the CF introduced in [6]:

- The *Building Cost* is the cost of the building PE capable of realizing a certain function. This cost is paid once independently of the number of times in which this PE is going to be called.
- The *Computational Cost* is considered to be the computing time required to perform a particular function. This cost has to be paid every time a PE is invoked or called by higher level PEs.
- The *Number of Calls* (NoCs) parameter stands for the number of times a processing element at a lower granularity level will be called to perform the task of the component at higher granularity level. A NoC is associated to every BF-reduction of every hyperarc.

We consider the following cost function:

$$\sum_i BC_i.N_i + \sum_n \sum_k CC_k((S_n)_{n\in\{1,2,\cdots,N\}}). \qquad (1)$$

where:

- $\sum_i BC_i.N_i$ is the total BC of all the PEs that are present in the multi-standard system, where $N_i \in \{0, 1\}$ depending on whether B_i^k is installed in the design or not.
- $\sum_k CC_k((S_n)_{n\in\{1,2,\cdots,N\}})$ stands for the total CC imposed by one of the N standards, S_n.
- $\sum_n \sum_k CC_k((S_n)_{n\in\{1,2,\cdots,N\}})$ is the total CC of all the N standards together.

An alternative theoretical expression of the above cost function can be found in [10].

B. An example for Cost Function derivation

In this subsection, we'll explain the computation process of the just introduced cost function expression through an example. In Fig. 4, the numerical values under each node will represent the building cost (BC)/ computational cost (CC) of the PE. For what follows, the term weight will be used to represent the NoCs. For instance, the NoCs (7) on the BF-reduction $(\{A\}, \{E\})$ of the hyperarc $(\{A\}, \{D, E, F\})$ in fig. 4 will be expressed as $w(\{A\}, \{E\}) = 7$.

Note that there are cases in which we have the same BF-reduction via different hyperarcs, as in Fig. 4 for example, we have $(\{A\}, \{D\})$ is a BF-reduction of both E_2 and E_3. Thus, it was necessary to specify the weight of each by adding the index which is the hyperarc from which it was derived. For the sake of simplicity, the BF-reduction $(\{A\}, \{D\})$ will be abbreviated by (A, D). Consequently, we will write: $w_{E_2}(A, D)$ (= 20 in Fig. 4) and $w_{E_3}(A, D)$

(= 15 in Fig. 4) for the weights of the BF-reduction (A, D) via E_2 and E_3 respectively.

In the following, we will compute the implementation cost of the S block at different levels of implementation.
As a first case, suppose that S is going to be installed by itself as a unique nondivisible block. Then the cost to be paid to perform the functionality of S will be:

$$\begin{aligned} Cost(S)_{(Using\ S)} &= BC(S) + CC(S) \\ &= 300 + 180 \\ &= 480. \end{aligned}$$

Now, suppose that a designer chooses to use a block of lower level than S (the A block) to implement the S functionality. Block A will be installed and it will be called $w_{E_1}(S, A)$ times (2 times) in order to perform the required tasks. Consequently, the cost becomes:

$$\begin{aligned} Cost(S)_{(Using\ A)} &= CC(A) \times w_{E_1}(S, A) + BC(A) \\ &= (20 \times 2) + 50 \\ &= 90. \end{aligned}$$

The task of S can be performed with the aid of components that are at even lower granularity level, lower than that of A. Here we have two options: either use B, D, & E or select the operators D, E, & F.
First we'll consider the former case. The total CC to perform the functionality of one A block using this option is:

$$\begin{aligned} CC(A)_{(Using\ B,D,E)} &= CC(B) \times w_{E_2}(A, B) + CC(D) \times \\ &\quad w_{E_2}(A, D) + CC(E) \times w_{E_2}(A, E) \\ &= 1 \times 30 + 2 \times 20 + 10 \times 5 \\ &= 30 + 40 + 50 \\ &= 120. \end{aligned}$$

Now since we need to call the tasks similar to the As tasks twice, then the CC that is going to be paid to realize S from

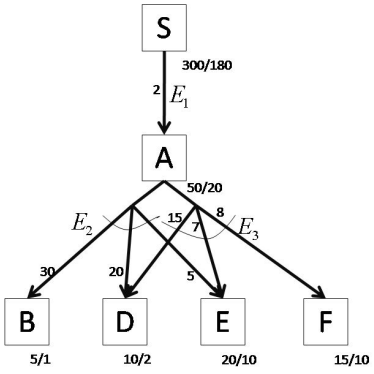

Fig. 4. A simple directed hypergraph showing the breakdown of block S up to 2 lower levels

B, D, & E operators will be:

$$\begin{aligned} CC(S)_{(Using\ B,D,E)} &= CC(A)_{(Using\ B,D,E)} \times w_{E_1}(S, A) \\ &= 120 \times 2 \\ &= 240. \end{aligned}$$

The final total cost of this choice will be to add the total CC of S (using B, D, & E) with the BC of each of B, D, & E only once. So:

$$\begin{aligned} Cost(S)_{(Using\ B,D,E)} &= CC(S)_{(Using\ B,D,E)} + BC(B) \\ &\quad + BC(D) + BC(E) \\ &= 240 + 5 + 10 + 20 \\ &= 275. \end{aligned}$$

Similarly, we'll calculate the cost of realizing S via D, E, & F (again according to equation 1) and we'll get:

$$\begin{aligned} Cost(S)_{(Using\ D,E,F)} &= (CC(D) \times w_{E_3}(A, D) + CC(E) \times \\ &\quad w_{E_3}(A, E) + CC(F) \times w_{E_3}(A, F)) \times \\ &\quad w_{E_1}(S, A) + BC(D) + BC(E) + \\ &\quad BC(F) \\ &= (2 \times 15 + 10 \times 7 + 10 \times 8) \times 2 + \\ &\quad 10 + 20 + 15 \\ &= (30 + 70 + 80) \times 2 + 45 \\ &= 360 + 45 \\ &= 405. \end{aligned}$$

The following summarizes all the attained costs to implement S:

$$\begin{aligned} Cost(S)_{(Using\ S)} &= 480 \\ Cost(S)_{(Using\ A)} &= 90. \\ Cost(S)_{(Using\ B,D,E)} &= 275. \\ Cost(S)_{(Using\ D,E,F)} &= 405. \end{aligned}$$

If for instance a designer seeks a least cost design to implement S, then he will be choosing the A operator to install inside the system because realizing S using the A operator yielded a minimum cost (90).

As you can see in this example, there were four options able to realize the functionalities of the S block. In the next section, our aim will be to provide a generalization for the total number of options capable of implementing the functionalities of the multi-standard system. In particular, we find an upper bound for this number, providing and explaining all the necessary derivations.

IV. An upper bound for the number of options of implementation

In this section, we will provide an exponential upper bound for the total number of options that are capable of implementing the multi-standard system. This is done by first introducing a computational cost vector X_v on each vertex v in the graph structure of the multi-standard system, whose dimension will represent the total number of options that can

realize block v. Then, we find an upper bound for $dim(X_v)$, thus achieving our goal.
Remark that one component in the X_v vector will represent the total computational cost of an option which is capable of realizing the functionalities of block v, where this cost will be calculated with the aid of the cost function introduced in the previous section.

A. Introducing the computational cost vector X_v

We'll associate a vector X_v to each vertex v in the directed hypergraph H of a multi-standard system. Each entry of X_v evaluates the total computational cost resulting from one particular choice of implementation chosen to realize block v. This vector will include all the possible implementations of v and thus the dimension of X_v will be exactly equal to the total number of options capable of realizing v.

For all the rest, we'll denote the dimension of the vector X_v by $|X_v|$ i.e $|X_v| = dim(X_v)$.
The parameters that we need to form the entries of the X_v vector are the BC, CC and the NoCs.

The vector X_v is defined recursively from blocks of lowest levels up until those of highest ones. This means that first we have to find X_v for all v block in level 1, then X_v for all v block in level 2, $\cdots$.
First, for blocks v in level 1, we have only one entry in X_v (i.e $|X_v| = 1$) because the only choice of implementation of such a block is by installing it itself. This only entry in X_v will be the CC of block v, because this will be the total CC in case where the block is installed by itself.

Having defined X_v for all the blocks v such that $l(v) \preceq i$, we can find the vector X_v where $l(v) = i + 1$ as follows:

- If we face an "or" hyperarc $e \in FS(v)$ (recall that an "or" hyperarc means that $|H(e)| = 1$) and suppose that $H(e) = \{r\}$ (so $e = (T(e), H(e)) = (\{v\}, \{r\})$), then this means that v can be realized by r associated with certain number of calls. Since r can be implemented in $|X_r|$ ways, then this will impose $|X_r|$ choices capable of realizing v by using r because the total CC of any option that implements r, multiplied by the number of times v calls r ($w(v, r)$), represents the total CC of an option that realizes v. Thus, any entry in X_r multiplied by $w(v, r)$ will be an entry in X_v describing the total CC of one of the options of implementation of v via r.
- If an "and" hyperarc $e \in FS(v)$ is encountered, then v will need the functionalities of all the blocks in $H(e)$ to be implemented. Suppose that $H(e) = \{s_{i1}, s_{i2}, \cdots, s_{in}\}$ (so $e = (T(e), H(e)) = (\{v\}, \{s_{i1}, s_{i2}, \cdots, s_{in}\})$). In this case, the calculation of the total computational cost of an option chosen to implement v via this hyperarc e will be:
  choose one entry from each of $X_{s_{ik}}$, $k \in \{1, 2, \cdots, n\}$ (which represents the total CC of one of the options that can realize the functionalities of s_{ik}), multiply it by the number of times v calls s_{ik} (which is $w(v, s_{ik})$), and then add all the answers obtained for all $k \in \{1, 2, \cdots, n\}$. This will form an entry of X_v, because this is the way how the total CC of an option that can realize v will be calculated when facing an "and" connection (as indicated in the previous section). Consequently, it's obvious that this hyperarc imposes $|X_{s_{i1}}| \times |X_{s_{i2}}| \times \cdots \times |X_{s_{in}}|$ options capable of realizing v.
- One more option of implementation which is worth mentioning is characterized by using v itself as a unified nondivisible block. This adds one more entry to the vector X_v which is the CC of block v.

To make things clearer, we'll start with a simple example and then introduce a generalization.

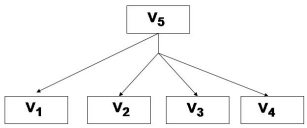

Fig. 5. two alternatives to implement v_5

1) Example: Fig. 5 shows an "or" ($e_1 = (\{v_5\}, \{v_1\})$) and an "and" ($e_2 = (\{v_5\}, \{v_2, v_3, v_4\})$) connection related to the implementation of block v_5.
Suppose that:

$$\begin{array}{c} X_{v_1} = (x_1, x_2, \cdots, x_m) \quad ; |X_{v_1}| = m \\ X_{v_2} = (y_1, y_2, \cdots, y_n) \quad ; |X_{v_2}| = n \\ X_{v_3} = (z_1, z_2, \cdots, z_p) \quad ; |X_{v_3}| = p \\ X_{v_4} = (l_1, l_2, \cdots, l_q) \quad ; |X_{v_4}| = q \end{array}$$

We'll denote Z_v by the set obtained from X_v by just listing its components. So, for example, we have $Z_{v_1} = \{x_1, x_2, \cdots, x_m\}$, $Z_{v_2} = \{y_1, y_2, \cdots, y_n\}$, $\cdots$.

Set $U_{e_1} = Z_{v_1}$ (related to the "or" connection)
and $U_{e_2} = Z_{v_2} \times Z_{v_3} \times Z_{v_4}$ (concerning the "and" connection).

In this case, X_{v_5} will be a vector of dimension $m + n \times p \times q + 1$ where:

- The m entries result from U_{e_1} as follows:
  $w(v_5, v_1) \times x_i \ \forall i = 1, \cdots, m$.
- The $n \times p \times q$ entries which result from U_{e_2} are:
  $w(v_5, v_2) \times a + w(v_5, v_3) \times b + w(v_5, v_4) \times c$
  $\forall (a, b, c) \ \in U_{e_2}$ (where $|U_{e_2}| = n \times p \times q$)
- The remaining entry is the result of the direct implementation of block v_5 itself. This entry, as mentioned before, is equal to the CC of block v_5.

2) Generalization: Let H be a directed hypergraph of a multi-standard system. Let $v \in V(H)$.
The components of X_v will be found as follows:
Let $e \in FS(v)$. Set $U_e = \prod_{r \in H(e)} Z_r$

$\forall e \in FS(v)$, $\forall a = (a_r)_{r\in H(e)} \in U_e$ we have: $\sum_{r\in H(e)} w(v,r).a_r$ is an entry in X_v.
One more entry of X_v is the computational cost of v which, as mentioned previously, presents the direct installation of block v.

B. An upper bound for $|X_v|$

1) The dimension of X_v, $|X_v|$: Based on all what's explained and discussed before in this section, we can easily conclude that :

$$|X_v| = \sum_{e\in FS(v)} \prod_{r\in H(e)} |Z_r| \; + \; 1 \tag{2}$$

defined recursively from lowest to highest levels.

For all the following, we'll denote u_i by an upper bound for $|X_v|$, where v occupies the i^{th} level i.e:

$$\forall\, v \;/\; l(v) = i, \quad |X_v| \leq u_i$$

An expression of u_i will be our desired upper bound. u_i will be, as well, defined recursively from lowest to highest levels.
The following two parameters will be used:

$$M = \max_{v\in V(H)} (|FS(v)| + 1)$$
$$r = \max_{e\in E(H)} (|H(e)|)$$

Taking into consideration equation. 2, we obtain recursively the following:

$$u_1 = 1(obvious)$$
$$u_2 = M(u_1)^r = M$$
$$u_3 = M.(u_2)^r = M(M)^r = M^{r+1}$$
$$u_4 = M.(u_3)^r = M(M^{r+1})^r = M^{r^2+r+1} \dots$$

So generally, we can express the recursive relation as follows:

$$\begin{aligned} &u_1 = 1, \\ &u_{s+1} = M(u_s)^r \quad \forall\, s \geq 1 \end{aligned} \tag{3}$$

Here is a brief explanation. Suppose that we want to find u_s, which is an upper bound for the number of options that can realize v (equivalently represented by $|X_v|$) where $l(v) = s$. We consider the following remarks:

- We have a maximum of $M-1$ hyperarcs such that v is the parent tail node (by the definition of M).
- Each one of these hyperarcs contains a maximum of r head nodes.
- The worst case is that all the r head nodes are in level $s-1$, which will impose a larger upper bound.

We can easily conclude, from equation 2, that each one of these (maximum) $M-1$ hyperarcs imposes a maximum of $(u_{s-1})^r$ options. And thus the $M-1$ hyperarcs all together will yield a maximum of $(M-1)(u_{s-1})^r$ options. It remains to add the last option of implementation characterized by using the block v by itself. So as a whole, we get a maximum of $(M-1)(u_{s-1})^r + 1$ alternatives of implementation which is obviously less than $M(u_{s-1})^r = u_s$ $(M \geq 1)$.

This recursive relation can be easily solved with a simple induction on s and we obtain: $u_s = M^{r^{s-2}+r^{s-3}+\dots+r+1}$ or alternatively:

$$u_s = M^{\frac{1-r^{s-1}}{1-r}} \tag{4}$$

In this section, we have found an exponential upper bound, as function of the selected parameters M and r, for the total number of options that can implement any PE in the design. Consequently, we can derive an upper bound for the total number of options of implementation of all the top level standards to be supported, and thus have explored the number of alternatives that can realize the multi-standard design.

V. Conclusion

After having theoretically described the graph structure of a multi-standard system as a directed hypergraph and introduced a cost function equation, we have provided an exponential upper bound for the total number of options which can implement the multi-standard system. This is an important step that might help in solving our optimization problem, which states to select the option capable of implementing the design and which has a minimum cost. This could be a clue towards discovering the complexity of this optimization problem, i.e whether it's a polynomial (P) or Non-deterministic Polynomial (NP) problem, which is an issue to be addressed in the future. Moreover, trying to improve this upper bound by finding a smaller one might be a very beneficial aspect to be explored.

References

[1] F.Jondral, "Parametrization: A technique for SDR implementation," Chapter 8 of Software Defined Radio Enabling Technologies, edited by W. Tuttlebee, Wiley, 2002.

[2] A. Kountouris, C. Moy, L. Rambaud, "Reconfigurability: a Key Property in Software Radio Systems," *in Proceedings of 1^{st} Karlsruhe Workshop on Software Radios*, Germany, March 2000.

[3] A. Kountouris, C. Moy, "Reconfiguration in Software Radio Systems," *in Proceedings of 2^{nd} Karlsruhe Workshop on Software Radios*, Germany, March 2002.

[4] Alaus L, Palicot J, Roland C, Louet Y and Noguet D, Promising Technique of Parameterization For Recongurable Radio, the Common Operators Technique : Fundamentals and Examples, march 2009, Revue Journal of Signal Processing Systems, diteur Springer New York.

[5] Moy C, Palicot J, Rodriguez V, Giri D, "Optimal Determination of Common Operators for Multi-Standard Software-Defined Radio", 4^{th} Karlsruhe Workshop on Software Radios, March 2006.

[6] Sufi T. Gul: Optimization of Multi-Standards Software Defined Radio Equipments : A Common Operators Approach, PhD thesis, 2009.

[7] Patricia Kaiser, Sufi Tabassum Gul, Christophe Moy, Yves Louet "Graph theory approach for optimization of Multi-standard software Defined Radio equipments", ERRT 6th karlsruhe, Mainz, Germany, June 2010.

[8] Reinhard Diestel. Graph Theory. Fourth Edition 2010. Springer-Verlag, Heidelberg Graduate Texts in Mathematics, Volume 173.

[9] G. Gallo, G. Longo, S. Pallottino, and Sang Nguyen. Directed hypergraphs and applications. Discrete applied mathematics, 1993.

[10] Patricia Kaiser, Yves Louet, Amine El Sahili "A cost function expression for SDR multi-standard systems design using directed hypergraphs ", URSI 2011, August 2011, Istanbul, Turkey.